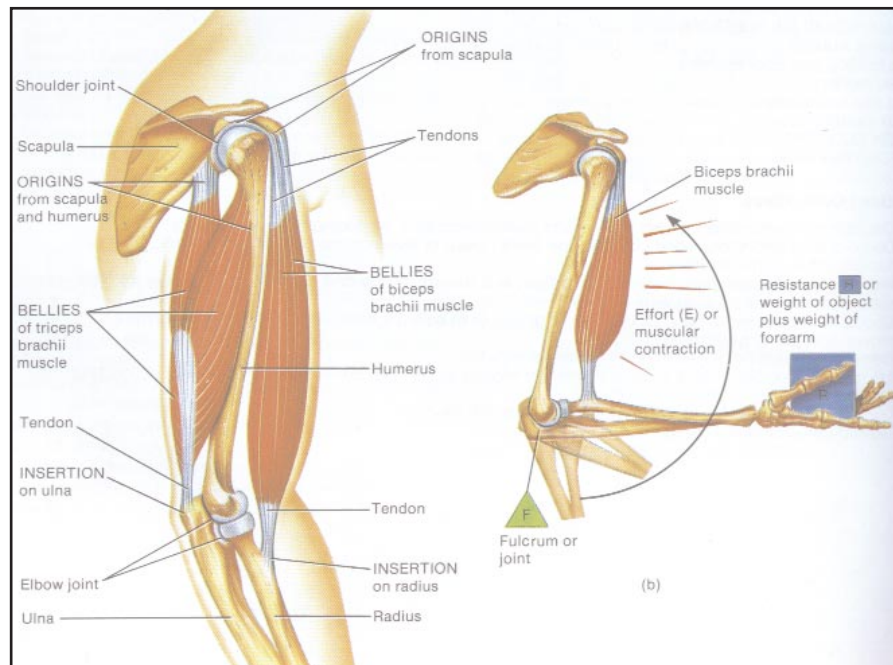


Gridpoints

The Quarterly Publication of the Numerical
Aerospace Simulation Systems Division



Researchers at Stanford University, supported by NASA and the NAS Systems Division, are studying muscle control to determine how it produces dexterity and its implications for future NASA missions. Page 6

FEATURES

- **New NAS Applications Branch Formed — 4**
- **IPG Team Meets Large Scale Computing Milestone — 10**
- **ILab: New Tool For IPG Parameter Studies — 12**
- **Computers For Future Space Missions — 15**
- **New Version of Chimera Grid Tools Released — 18**

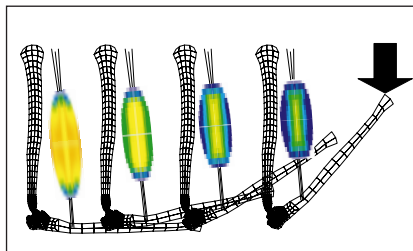
Features

6

Muscle Modeling: The Quest For Dexterity

Researchers at Stanford University, supported by NASA and the NAS Systems Division, are studying muscle control to determine how it produces dexterity, and its implications for future NASA missions.

Nicholas A. Veronico

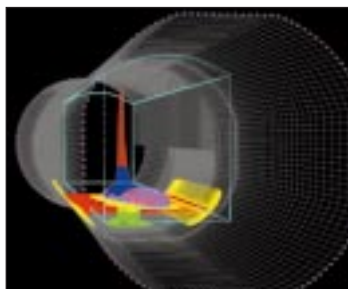


10

IPG Team Meets Large Scale Computing Milestone

To meet its latest milestone, the Information Power Grid team incorporated the updated 512-processor Origin 2000, and ran a 14-million point computation across three remotely located computers.

Holly A. Amundson



12

ILab: New IPG Tool Improves Parameter Study Implementation

NAS researchers have designed a tool to fully automate parameter study creation and launching within the IPG.

Holly A. Amundson

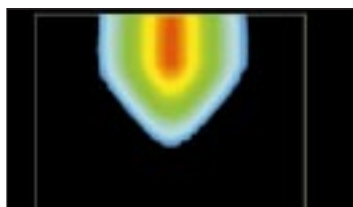


15

Space Computing – The Present Frontier

Computer modernization and miniaturization hold the key to longer-duration space missions to the outer reaches of the solar system.

Holly A. Amundson



18

Chimera Grid Tools Software

A new, updated version of the comprehensive Chimera Grid Tools software package for over-set grid generation has recently been released.

William M. Chan and Stuart E. Rogers



On The Cover:

Researchers are studying muscles as a distributed control system. As a person lifts an object in their hand, the biceps muscle flexes to produce a resultant movement around the elbow. See page 6 for the full story. (G.J. Totor's Principles of Anatomy and Physiology, Sixth Edition, used with permission of John Wiley & Sons, Inc.)

Gridpoints

The Quarterly Publication of the Numerical Aerospace Simulation Systems Division

Vol. 2, No. 1 • Spring 2001

Executive Editors: Bill Feiereisen
John Ziebarth

Editor: Nicholas A. Veronico

Writers: Holly A. Amundson
William M. Chan
Jill A. Dunbar
Stuart E. Rogers

Graphic Designer: Sue Kim

Illustrator: Gail Felchle

Contact the Editor:
Nicholas A. Veronico
Gridpoints
M/S 258-6
Moffett Field, CA 94035-1000
Tel. (650) 604-3046
Fax (650) 604-4377
nveronico@mail.arc.nasa.gov

The staff of *Gridpoints* would like to thank the following people for their assistance in preparing this publication: Michael Aigner, Dick Anderson, Michael Boswell, Dan Cooke, Christine Cortez, Glenn Deardorff, M. Jahed Djomehri, Chris Gong, Ana Grady, T.R. Govindan, Jean H. Heegaard, Bill Johnston, Chris Kleiber, Karen McCann, Nagi N. Mansour, Shelley Marshall, Meyya Meyappan, Gina Morello, Paul Munoz, Todd Novak, Marcia Redmond, Frank Sabadin, Ryan Spaulding, Valerie Steele, Ron Sykora, Leigh Ann Tanner, Bill Thigpen, Arsi Vaziri, Cliff Williams, Maurice Yarrow, Steve Zornetzer

Gridpoints is a quarterly publication of the NAS Systems Division at NASA Ames Research Center. The division's staff includes government employees and contractors from Advanced Management Technology Inc., Computer Sciences Corp., Cray Inc., Eloret, Raytheon Company, and SGI.

Gridpoints acknowledges that some words, model names and designations mentioned herein are the property of the trademark holder. *Gridpoints* uses them for identification purposes only.

Subscribe to *Gridpoints*

Subscriptions to *Gridpoints* are complementary. Please send e-mail with your name, company name, title, address, and zip code to:
gridpoints@nas.nasa.gov
or subscribe online at:
www.nas.nasa.gov/gridpoints/subscribeform.html

Note: Subscribers outside the continental United States can access *Gridpoints* on the web in PDF format at: <http://www.nas.nasa.gov/gridpoints>

News From NAS

New NAS Research Branch Chief Appointed

NAS Systems Division Chief Bill Feiereisen announced recently that Bryan Biegel has been appointed chief of the division's Research Branch. The branch consists of some 40 programmers and scientists whose work encompasses computer science research and development for aerospace applications, NASA's Information Power Grid (IPG), and nanotechnology, among other areas.

"The challenges and possibilities of the Research Branch Chief position are compelling to me," Biegel said. His first order of business is to ensure that the branch technical staff can work with minimal bureaucratic "red tape," and that the staff's talents and expertise are fully utilized and acknowledged. "I'm very excited about helping everyone in the branch to work happily and efficiently, so that we will in turn help NAS continue to do great things in high-performance computing."

Biegel has worked in the NAS organization for more than three years as a researcher in the semiconductor device modeling group. Before accepting the

Continued on Page 2

NAS Mission

To lead the country in the research and development of high-performance computing for NASA Programs and Missions by being the first to develop, implement, and integrate new high-performance computing technologies into useful production systems.

To provide NASA and its customers with the most powerful, reliable, and usable production high-performance computing systems in the country.

From The Division Chief

Almost three years ago, the NAS Division began NASA's Information Power Grid (IPG) project at a time when the concept of Distributed High Performance Computing was in its infancy. The division joined an international community that had come together out of mutual interest to develop a common compute and data infrastructure for science and engineering. The grid community evolved from a vision of shared access to resources, and defined a research agenda to achieve this vision. At that time, there was no running infrastructure that could be provided to scientists and engineers; however, there was a shared view of the research areas that needed to be addressed in order to build the grid.

The grid project has come a long way in three short years. The once casual grid community has evolved into a formal organization known as the Grid Forum. This group has fostered world-wide interest and participation in defining the grid. Various national and international grid projects have begun. I'm pleased that NASA and NAS – along with our National Science Foundation (NSF)/Partnership for Advanced Computational Infrastructure (PACI) and Department of Energy colleagues – are on the leading edge of this revolution in the U.S., and are actively becoming engaged with our overseas colleagues.



I'm writing this while participating in the first Global Grid Forum (<http://www.ggf1.nl>), a working meeting where technical experts are defining the standards upon which all the international grid projects will interoperate. The first Grid Forum, held two years ago, was hosted by NAS. (We were worried that we could not accommodate the 110 participants in our auditorium.) The Global

Grid Forum has advanced far beyond our humble beginnings. Attendance this year had to be capped at 325 participants — remember, this is a working meeting, and we had to turn people away. To have such interest in the creation of our own computing future is quite gratifying.

One of NASA's great strengths is to carry research tasks through to working systems. This has been the IPG team's focus for the last three years – to put in place an infrastructure to provide researchers and engineers the computing power to advance science. A working grid now exists that includes high-performance systems at three NASA research centers. In addition, we are about to link IPG with the grids that our colleagues are building at the NSF/PACI centers. Uniform access to many disparate resources has been touted as IPG's advantage, but ease of use has lagged. Portals have sprung up in order to make this access easier and more uniform, and I invite you to visit the Information Power Grid portal, the *IPG Launch Pad* at <http://www.ipg.nasa.gov> and see what this is all about. All of this grid work remains focused on supporting the science and engineering of NASA's missions. I hope you enjoy the articles in this issue, which feature some of the exciting work that will increasingly be done by using the grid as a tool.


As always, I welcome your feedback.

Bill Feiereisen

wfeiereisen@mail.arc.nasa.gov

News From NAS

branch chief position, he was employed by Computer Sciences Corp. under a NASA contract. His most recent research projects have focused on quantum effects in ultra-small conventional electronic devices, thermal and radiation effects in space electronics, and electronic device modeling in the IPG environment.

Prior to joining the NAS division in 1997, Biegel was a research assistant at Stanford University, where he developed SQUADS, the Stanford Quantum Device Simulator, an automated, extensible software package for accurately modeling one-dimensional quantum electronic devices. Biegel holds both a bachelor of science degree and master's degree in electrical engineering from the University of Oklahoma, and earned his Ph.D. in electrical engineering from Stanford. 



Bryan Biegel

High Dependability Computing Workshop a Success


The first High Dependability Computing Consortium (HDCC) workshop was held at the NAS Facility January 10-12 to gather parties interested in making computer software more reliable for the future. Partnering to form the HDCC, Carnegie Mellon University and NASA Ames Research Center had the same goal in mind – to eliminate computer failures.

“This is an extremely important event for NASA – future missions depend on the long-term success of software reliability and dependability,” says Steve Zornetzer, director of Information Sciences and Technology at Ames. “This consortium represents a very unique and bold approach to tackling one of the most vexing problems, and potentially one of the most important problems for the advancement of information technology in the 21st century,” he says.

When a computer fails, any number of things can go wrong – downtime results in hundreds of thousands of e-commerce dollars lost every hour, computer failures in space endanger NASA mission success, and computer software failures at the bank translate to no money at the ATM. All these problems can be avoided by funneling time and effort into ensuring dependability of the software used to run computers.

During the three-day event, participants from industry, government, and academia reviewed case studies and participated in technology discussions to start the wheels turning on the journey to software reliability. Speakers at the workshop based their presentations on past software failures, present capabilities, and requirements for future computing to emphasize the importance of software dependability. Participants included specialists from the University of

Southern California, Hewlett-Packard, NASA Goddard Space Flight Center, Bell Laboratories, the Department of Defense, and Canada's McMaster University, among others.

“I'm hoping that this will be the beginning of a continuous process by which we put together a long-term effort (20 to 50 years) to build not just the industry, but the field of computer science,” says Jim Morris, dean of the School of Computer Science at Carnegie Mellon University. The workshop planning committee will schedule more events in the coming months. 


New Web Features for NAS Users

A new, secure tool for scientists using the NAS Facility's supercomputing resources debuted January 23 on the division's website. The System Status page allows users to view important information on the operational status and utilization of computing and storage resources. Users can also check scheduled maintenance periods and get other useful systems data, which is updated automatically every 10 minutes.

From one page, users can: get the message of the day (MOTD) for each user type; see the schedule for upcoming system outages – and review previous outages; find out if any systems are “down;” check system status history; view network router status; as well as view and print a histogram showing system and storage utilization.

Local NAS Facility users can also get status on printer queues. The System Status page was developed by Ernst Kimler in the NAS Engineering Branch.

Also new to the User Services section: a User FAQ page, with answers to basic questions about using NAS computing resources, and an easy-to-access user help form to request assistance with a problem.

For user-related assistance, contact NAS User Services at nashelp@nas.nasa.gov or call (800) 331-8737. 

NAS Scientist Named to Editorial Post


NAS Systems Division senior scientist Cun-Zheng Ning has been named associate editor of the *Journal of Quantum Electronics* (JQE), the monthly journal of the IEEE-LEOS (Lasers and Electro-Optics Society). Ning's appointment was unanimously approved by the IEEE-LEOS Board of Governors.

During his three-year tenure, which began in January 2001, Ning's duties will include identifying reviewers and making decisions regarding acceptance of submitted papers. He has published extensively in the fields of optoelectronics, lasers, and nonlinear physics, and has served as conference chair or member of technical committees in the field of optoelec-

tronics at many international conferences. Ning is task lead for device modeling projects within the NAS division. He obtained his Ph.D. in theoretical physics at the University of Stuttgart, Germany. He was a research assistant professor at the University of Arizona before joining NAS. Ning is employed by Computer Sciences Corp.



Cun-Zheng Ning

IEEE-LEOS is considered to be the one of the most influential organizations worldwide in the fields of lasers, optoelectronics, and optics sciences and engineering. Since inception in the early 1960s, JQE has consistently received one of the highest citation ratings of any IEEE journal. 

New Version of PEGASUS Code Released

A public beta release of the PEGASUS code (version 5.1) is now available. PEGASUS is a key component in a series of tool used in computational fluid dynamics (CFD) analysis. Version 5.1 significantly reduces user effort and allows for easier operation of the code.

The PEGASUS code is responsible for connecting overset volume grids and preparing them for the OVERFLOW flow solver. The new version dramatically reduces the time required to perform the overset connection. The result is a process which is automatic for many flow analysis problems.

The PEGASUS code, developed at NASA Ames Research Center in collaboration with Microcraft Corp., is used by both government and industry users. The code is used in conjunction with the Chimera Grid Tools (used for grid generation) and OVERFLOW. A parallel version of PEGASUS is in the works.

Interested readers can contact NAS researcher Stuart Rogers at rogers@nas.nasa.gov for additional details. To obtain additional information on PEGASUS, including a software request form and user's manual, visit: <http://www.nas.nasa.gov/%7Erogers/pegasus/intro.html>

Mars Landing Site Archive Now Online

An extensive archive of Mars data is now available online to assist in the selection of landing sites for the Mars Explorer Rover (MER) twin rover missions scheduled for launch in 2003. This interactive website, called the Marsoweb (<http://marsoweb.nas.nasa.gov/landingsites/>) is a collaboration between the Center for Mars Exploration (CMEX) and the NAS Systems division. CMEX principal investigator Virginia Gulick provides general guidance and

NAS Technical Seminars Span Range of Topics

On February 26 2001, Nirav Kapadia was the guest speaker at a NAS Information Power Grid Seminar where he presented the talk, "Integrating Grid services and Web Technologies to Power Application Portals: The PUNCH Approach." PUNCH (the Purdue University Network Computing Hubs) is a prototype system that allows users to run unmodified software via standard web browsers. Kapadia's focus was on Purdue's approach to key issues in the design of such a system, including: the interface to the external world; internal system design; support for legacy applications; and resource management across administrative domains.


On March 3, 2001, Robert Horst from 3ware, Inc. discussed several different approaches for building Storage Area Networks (SANs) using standard Ethernet switches and interfaces in his NAS New Technology Seminar "Ethernet Storage Area Networks." A taxonomy for the various types of network-attached storage was suggested, with the focus of the talk on the hardware and software technology of the native E-SAN implementation recently introduced in the 3ware Network Storage Unit.

These highlights are just two of the NAS technical seminars held recently. For a complete listing of NAS seminars, see the NAS technical training event Website : <http://www.nas.nasa.gov/User/Training/training.html> Videotapes of most seminars can be borrowed by sending your request to: doc-center@nas.nasa.gov. For questions, contact Marcia Redmond at: mredmond@mail.arc.nasa.gov



serves as contact to the Mars research community for data content, and NAS senior research engineer Glenn Deardorff is responsible for technical development. The pages contain data from both the current Mars Global Surveyor (MGS) and previous Viking Missions.

The landing site section of the Marsoweb for the 2003 missions was developed, in part, for a Mars landing site workshop held at NASA Ames, January 24-26. Graphics from the Mars site were used frequently in workshop participants' talks, according to Deardorff. "The website was well received and is being very actively used," he says.

The new landing site section enables researchers to navigate landing sites within the equatorial landing zone of Mars using Java-based image maps generated from geology maps, elevation data, high-resolution MGS images, and a number of other datasets. Much of the data available on the site was presented to the Mars research community for the first time at the January workshop. Deardorff has plans to incorporate datasets for the entire surface of Mars in the future. 

New Applications Branch Tackles Diverse Challenges

Believing in the power of individuals, and in harnessing that power to accomplish important goals for NASA, scientist Dochan Kwak is bringing researchers from diverse disciplines together in a new organizational branch. The branch is chartered to develop high-performance computing applications in space science, Earth and biological sciences, and the aerosciences applications that have always been a cornerstone of the NAS organization.

Complementing NAS's existing Engineering and Research branches, the Applications Branch completes the integration plan envisioned by Division Chief Bill Feiereisen. "If you think in terms of the various layers of technology, you start with hardware support, and above that is systems software, with tools above that, and applications at the top," says Feiereisen. (See chart below.) "Because NAS has always been dependent on people from other organizations to develop the applications, the idea was to bring in scientists from Code A (the Aerospace Directorate at Ames Research Center) to fill in the top of the 'vertical integration stack.'"

Because applications work can't be completely self-contained within one organization, adds Feiereisen, the new branch is additionally chartered to become a "conduit to other application areas," such as Earth sciences.

The former Code A scientists, who joined NAS in May 2000, have extensive experience in aerospace applications, and have close associations with NASA programs involved in

space transportation development and aerospace vehicles. "We are in contact with the aerospace vehicle developers and designers," says Kwak, who was previously a Code A branch chief. "So, while NAS's high-end computing has been designed to address some of the large-scale computational issues, now we (at NAS) can go a step further and demonstrate that these application tools and procedures are useful to NASA enterprises and industry customers."

Focus on Advanced Technology Applications

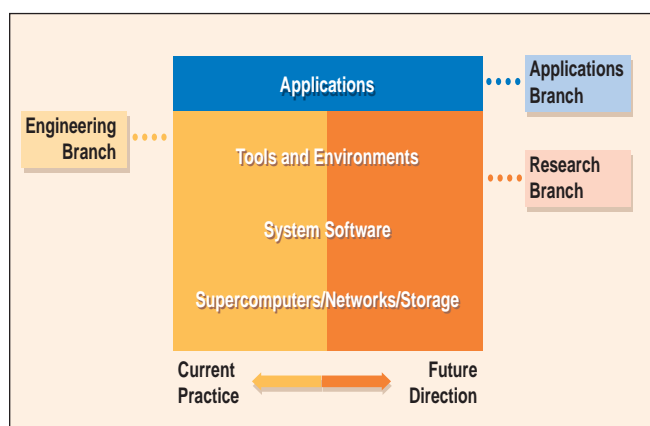
Since May 2000, the group has been working together to create the branch organization, charter, and goals. Up to now, and continuing through fiscal year 2001, Kwak explains, most branch members have been engaged in work on NASA's Advanced Technology Applications (<http://cas.arc.nasa.gov/grand.html>). ATA focuses on areas critical to the entire lifecycle of aerospace vehicles – from concept to operation – in order to reduce design cycle time and improve the safety and efficiency of future aerospace vehicles and systems.

One important NASA project, for example, is to develop tools and procedures applicable to second- and third-generation reusable launch vehicle activities, which require sophisticated computational tools and vast amounts of computing power. One of Kwak's teams is working with engineers at NASA's Marshall Space Flight Center on rocket subsystem simulations in order to develop engines for future reusable launch vehicles.

Another team is working on "powered lift vehicle simulation" associated with such high-performance aircraft as the vertical take-off Harrier jet in "ground effect" (when the vehicle is close to the ground, such as during landing). The team is demonstrating the use of high-fidelity computational fluid dynamics (CFD) and neural networks to generate a stability and control database using the Information Power Grid.

Beyond aerospace applications, branch members have expertise in developing computational methods and algorithms that can be applied to other sciences. "We can contribute to other areas such as biocomputing, for example in the area of hemodynamics (the movement of blood and the forces involved in blood circulation) and electromagnetic wave propagation in the human body," explains Kwak.

The group's work in hemodynamics includes manufactured devices such as the DeBakey Ventricular Assist Device, a



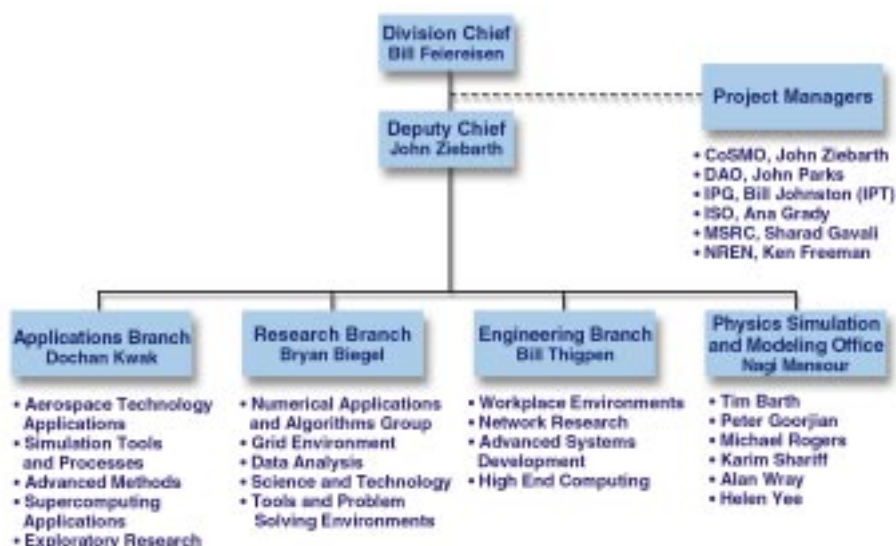
The NAS 'vertical integration stack' shows the types of technologies encompassed in the NAS mission, and where each branch in the division meshes with the others. The new Applications Branch spans both current and future technologies.

miniature heart pump. Kwak and NAS colleague Cetin Kiris employed computational fluid dynamics techniques and the division's high-performance computers to improve the blood flow through the life-saving device. (See below, *The Kind of Scientist We Need*.)

Creating Applications for the Grid

Another important aspect of the applications work is NASA's Information Power Grid (IPG) testbed, the distributed high-speed computing infrastructure being developed collaboratively by government, university, and industry partners. Feiereisen explains that the intent is to make the grid a platform for applications and tools' developers to work on. "I frequently say that the IPG is the focus of the division. So, I expect that everyone in the new branch – applications people, those doing CFD, biology people – will write (tools) for the IPG structure. The underlying series of platforms will be IPG-like, and they have to think in terms of developing their science and engineering for the future based on those platforms."

Kwak would like to use the technology developed under the ATA to demonstrate applications on the IPG in the not-too-distant future. "The IPG project team has made great progress to date. The next step is to make an actual impact



NAS's top-level 'org' chart shows the division's three branches: Applications, Engineering, and Research, as well as the Physics Simulation and Modeling Office. An underlying matrix organization dovetails with the three branches. This organizational structure fosters synergy and innovation, and provides flexibility for sharing people resources among the working groups. Bulleted items denote key areas or personnel for each branch.

on some significant NASA project. This group will add that element into the IPG work." — Jill A. Dunbar

Advanced Technology Applications are funded by the Computational Aerospace Sciences portion of NASA's High Performance Computing and Communications (HPCC) Program.

'The Kind of Scientist We Need'

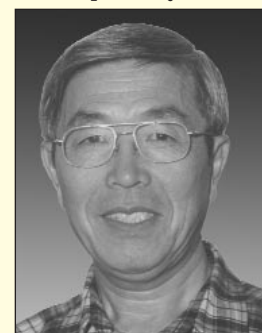
Dochan Kwak, newly appointed chief of the NAS applications branch, is a perfect fit for his new role. "He has broad experience in computational fluid dynamics (CFD), and by extrapolation, computational physics," says NAS Division Chief Bill Feiereisen. "Dochan has the expertise, experience, and management talent – he is just the kind of scientist we need."

In Kwak's 22-year career at Ames Research Center, he has served as chief of both the Advanced Computational Methods and the Computational Physics and Simulations branches. Previously, he was a staff scientist at Los Alamos National Laboratory. He holds a bachelor of science degree in mechanical engineering from Seoul National University, and a Ph.D. in aeronautics and astronautics from Stanford.

"I believe that the individual contributor is the most valuable asset to a research organization," Kwak says. "And I like to foster an environment where people can be productive and excel." To that end, he has been working with counter-

parts Bryan Beigel and Bill Thigpen, who lead NAS's Research and Engineering branches, respectively. "Communication has increased quite a bit, and we've encouraged our groups to work closely together."

Kwak has received numerous honors, including the First NASA Software of the Year award, for developing INS3D, a CFD code used for solving steady-state and time varying flow problems. He was inducted into the Space Technology Hall of Fame in April 1999. Among Kwak's many achievements are a patent (shared with several inventors) for the DeBaKey Ventricular Assist Device (see www.nas.nasa.gov/Main/Features/2000/Winter/debakey.html), a miniature heart pump that prolongs the life of patients awaiting heart transplants. In addition, he has authored more than 80 papers, is a favorite on the international science lecture circuit, and has served on many scientific committees.



Dochan Kwak

Muscle Modeling: The Quest For Dexterity

Researchers at Stanford University, supported by NASA and the NAS Systems Division, are studying muscle control to determine how it produces dexterity, and its implications for future NASA missions.

Walking from the desk to the printer to retrieve a printed e-mail, or picking up a pot of coffee to pour the morning's first cup are two luxuries most people take for granted. The loss of one's mobility never enters a person's mind until a high-profile celebrity such as Christopher Reeve becomes paralyzed. In the year 2001, humankind is capable of sending space probes to distant planets, but making a muscle move smoothly through artificial stimulation has eluded the world's finest physicians and scientists.

Muscles are essentially a distributed control system where one tool (the brain) manages the input of information from sensors (nerves) and instructs actuators (the neurons inside the muscle) to contract or relax, thus producing movement. In an effort to advance the theory and science of distributed control systems, NASA's Center for Turbulence Research, supported in part by the NAS Systems Division, is conducting a muscle modeling study by Stanford University

doctoral student Michael Aigner and assistant professor of biomechanical engineering Jean Heegaard. To better understand the mechanics of a distributed control system, the researchers sought an existing example on which to base their modeling efforts. Muscles are natural examples of a distributed control system and, although how they work is now well understood, how they are controlled to produce dexterity has yet to be determined.

"The musculoskeletal system is an actual physical system that we can see and test in the lab," says Aigner. "Over the years, researchers have observed various actuation, or movement patterns and how these patterns change inside the muscle. For example, a weight lifter would like to have the muscle's action synchronized because synchronized actuation produces maximum force; however, a pianist wants his or her finger muscle actuation unsynchronized, which delivers maximum dexterity. This shows our inspiration for studying not just one or two control variables, but rather a distribution of them across the entire muscle."

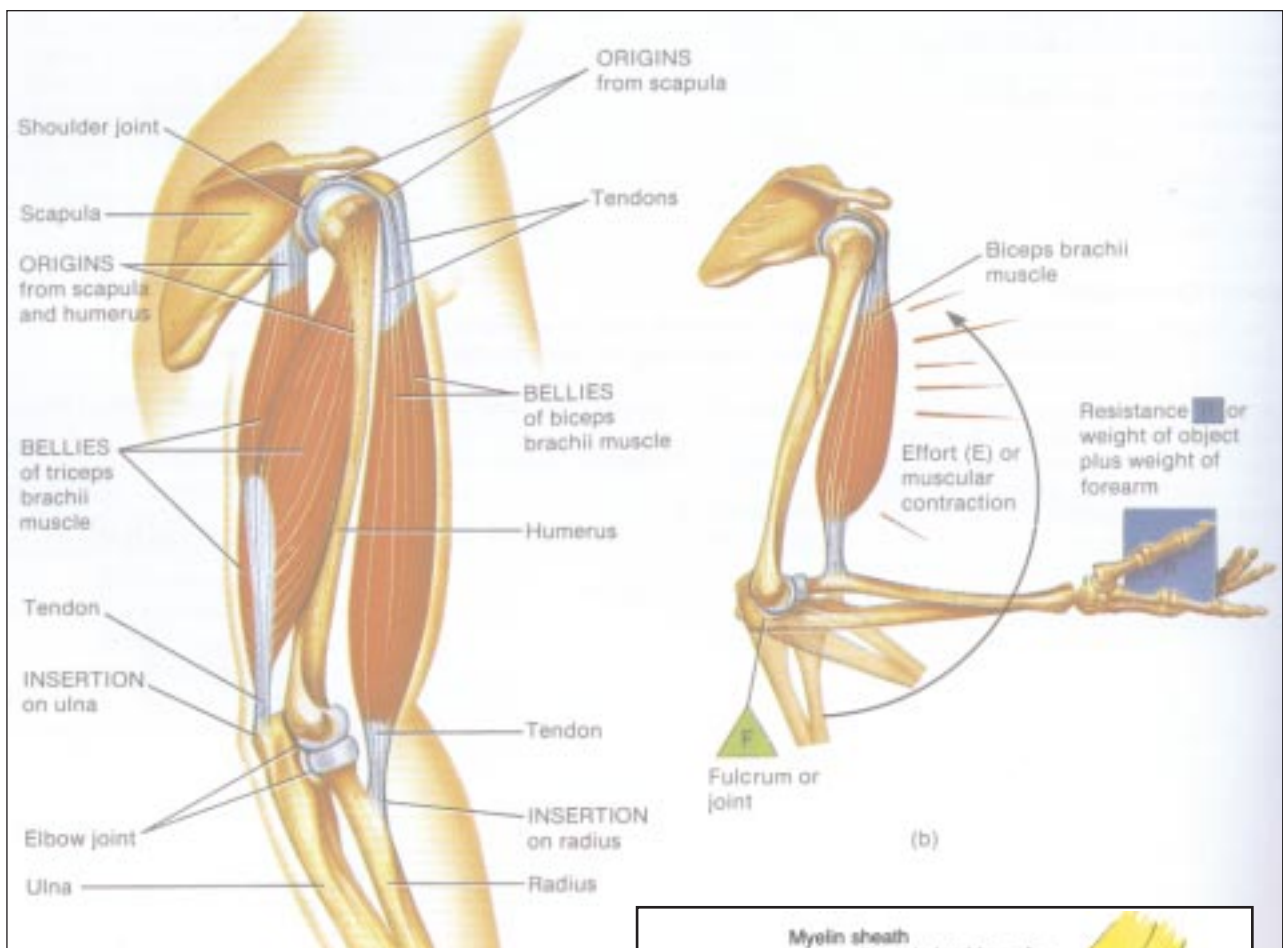
Center for Turbulence Research

The Center for Turbulence Research (CTR) is a joint NASA/Stanford center that sponsors the work of post-doctoral fellows on turbulence research. This includes studies of turbulence modeling, combustion modeling, turbulence control, aero-acoustics and, in general, research on flows where turbulence plays a dominant role. The center has a facility at Stanford and one at Ames Research Center (ARC). Co-located in the same building at ARC are civil service scientists from the Physics Simulation and Modeling Office of the NAS Systems Division who collaborate with CTR post-doctoral fellows and visitors. The distributed control work described in this article was motivated by CTR's work on distributed control for aerodynamic drag reduction.

Artificial Muscle Movement Today

Today's state of the art in the electrostimulation of muscles is represented by the "Stand Up and Walk" project at the University of Montpellier, France. Six patients recently received microcomputer implants (located in the abdomen), and electrode implants in the leg muscles. An external transmitter sends a signal to the implanted computer, which stimulates the leg muscles to expand and contract. Using this method, a paraplegic can lean on a walker and signal the computer to move the leg muscles to produce movement.

While Stand Up and Walk is an incredible accomplishment, the drawback is that through artificial stimulation, patients can only take about 10 steps before they are physically exhausted. "I've seen videos of paraplegic people using today's technology to walk in a lab for 10 yards, and then they're



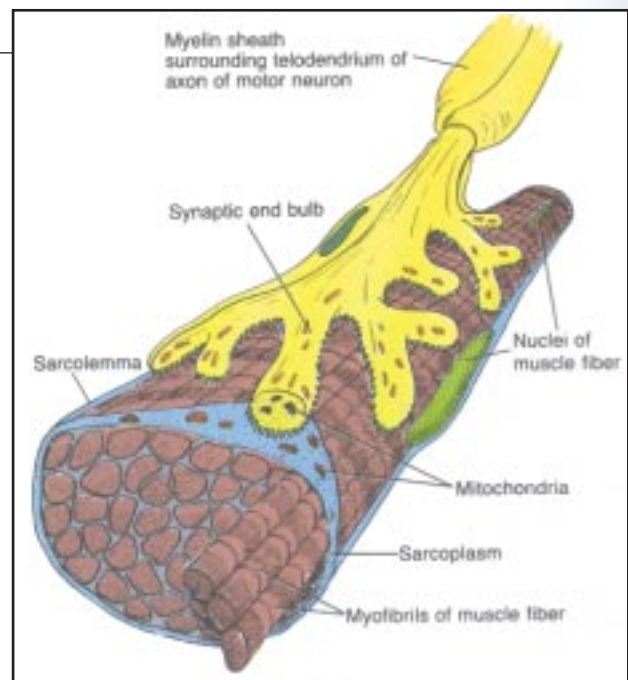
As a person lifts an object in their hand (above), the biceps muscle flexes to produce a resultant movement around the elbow. Zooming in on a microscopic level (right) distinguishes the structure of the myosin fibers and the attached neural network. (These figures are from G.J. Totor's Principles of Anatomy and Physiology, Sixth Edition [ISBN 0-471-367885], and are used with permission of John Wiley & Sons, Inc.)

totally exhausted because there is no dexterity in what they do,” says Heegaard. “They are walking like crazy, but its crazy that they’re walking at all! There is still so little understanding of what produces smooth motion.

“We must find out how muscles move using so little energy when controlled by the brain, versus why muscles require so much energy to perform the same task when artificially stimulated. So far it’s a brute-force method that is still very experimental. If we can understand the central nervous system pattern generator, one can start encoding it in small computers and have paraplegic people walk again. It’s far fetched, but conceptually it’s there.”

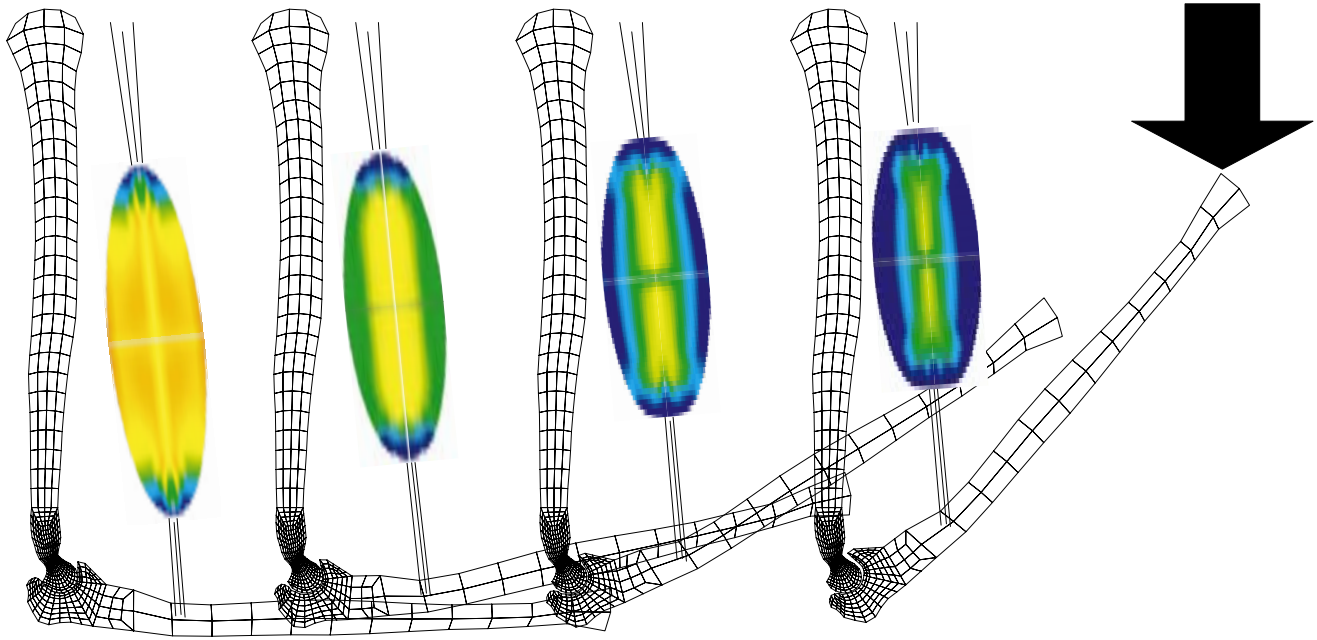
Computational Modeling

For muscle studies to advance, researchers must progress past today’s single-input parameter studies that assume there is only one activation variable per muscle. Using single-



input parameters, the contractile behavior of the entire muscle is controlled by a single neural signal. By contrast, distributed control muscle models will enable researchers to explore the dynamics that produce dexterity.

Aigner and Heegaard spent the early computational phase of their research (1999 to 2000) developing an understanding



This computational visualization depicts the bending of the elbow joint and lifting of the forearm through the contraction of the biceps muscle. An increasing load is applied at the wrist (black arrow) and the biceps muscle has to progressively contract to raise the forearm. The plots above represent the amount of hydrostatic pressure in the body of the muscle as a result of the force exerted on the tendons and the muscle's contraction. The blue shades represent tensile stress while the yellow depicts compressive stress. Notice the inhomogeneity of the stress distribution, with tensile stress at the periphery of the muscle and compressive stress in the bulk of the muscle belly as the forearm is raised. (Heegaard, Aigner; visualization by Eric Girod)

of how the muscle system behaved. Then the pair began examining the basic geometries of a two-dimensional muscle model and the physics involved in its movement. Aigner began the computational study using the computer program Matlab, and is now using "Tact," a custom finite element code written in part by Heegaard, to model more complex muscle systems. Tact is run on Linux-based PCs using dual Intel Pentium processors.

"We're typically dealing with problems that have anywhere from 1,000 to 50,000 degrees of freedom (or unknowns), and CPU times ranging from a few minutes to a few hours," says Heegaard. "We're not interested in having problems that are so large that it will take us weeks to run. From a research point of view, we want quick results to find out if what we're doing is working or not."

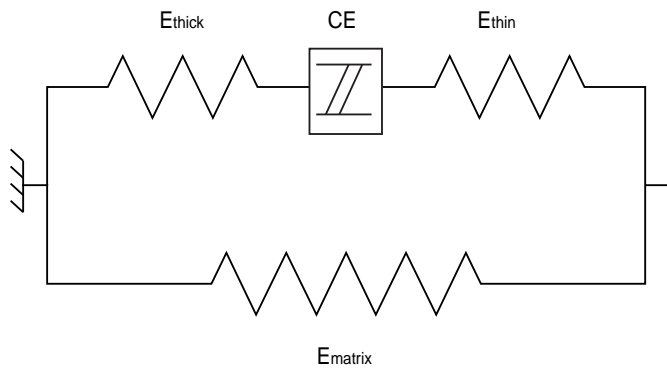
In the early 1990s, supercomputers at the NAS Facility were used by Clay Anderson, now at Stanford, to model the movement of the musculo-skeletal system. Anderson used the Intel Paragon supercomputer to predict the activation patterns on a series of 30 to 40 muscles. In Anderson's study, each muscle had only one control parameter, which was used to make a 3-D multi-body computer model of a skeleton walk. At that time, it took nearly four months to run the computation on an SGI workstation, and about a month on the Intel Paragon supercomputer. With advancements in

computer speed and technology over the past decade – depending upon the parameters, Aigner and Heegaard's computations can be run in a few hours – if not minutes – on their desktop computers.

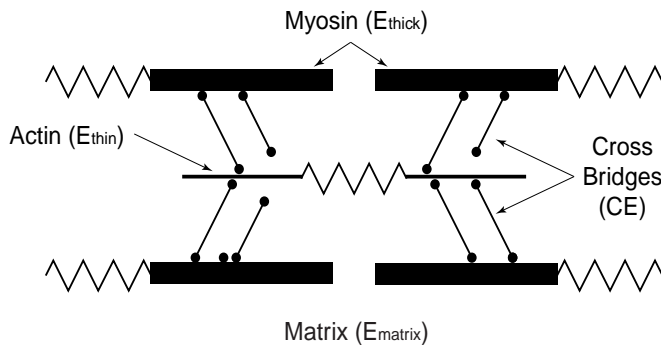
Aerodynamic and Robotics Implications

NASA is interested in the development of distributed control systems that have hundreds or thousands of actuators controlling the flow around a wing. The challenge is to develop a control system capable of managing the actuators and sensors while remaining light enough for flight. The fields of fluid dynamics and aero-elasticity have developed theories that revolve around the idea that hundreds or thousands of small flaps or spoilers on a wing could improve air flow over its surface. Aerodynamic drag is reduced by opening or closing the spoilers, depending on the aircraft's "nose up" angle in relation to its direction of flight.

"There is a parallel between a muscle and a wing system in the sense that the muscle is also made up of a large number of sensors and actuators," explains Heegaard. "The big question is, how do we deal numerically and computationally with the problem? How could one find the best configuration of flaps up or down? The physics of the problem is so complex that even if you had just a few parameters, calculating the movements of the control system would be difficult."




Above left: Rheological model for one-dimensional muscle contraction. The passive elasticity of the thick myosin (E_{thick}) and thin actin (E_{thin}) fibers plus that of the surrounding fibrous matrix (E_{matrix}) combine with the active behavior of the contractile element to generate muscle force.



Below left: The contractile element is a black box which represents the activation triggered ratcheting process of the cross-bridges between the myosin and actin. As more cross-bridges become attached, the greater the muscle's ability to voluntarily contract.

Studying distributed control systems also crosses over into the field of robotics. To give a robot the dexterity of a human hand, researchers must understand how the hand is controlled in order to mimic its actions. Describing the challenges of bringing dexterity to a robot, Heegaard says that while a simple robot may have five movement parameters, a muscle has a nearly endless amount of control inputs. Each input adjusts the muscle's movement slightly, thereby producing dexterity.

Future Simulations

Aigner and Heegaard's next step is to study a distributed muscle control system in action. As the project progresses, the researchers foresee modeling simulations of multiple muscles with distributed controls in a realistic musculo-skeletal model of a hand. "For that we will need very powerful computers and computational facilities, and that's where the NAS Systems Division comes in," Aigner said. "It will involve a whole new area of research and an entirely new set of computational challenges." 


—Nicholas A. Veronica

For more information on the Stand Up and Walk program:
<http://europa.eu.int/comm/research/success/en/med/0077e.html>

About The Researchers

Jean H. Heegaard is an Assistant Professor of Mechanical Engineering at Stanford University. He conducts research in the area of computer modeling of the human body with special focus on the musculo-skeletal system. Studies include the use of finite element, multibody dynamics and optimal control approaches to analyze joint biomechanics; evaluation of surgical procedures; and simulation of growth phenomena. He is a member of the Orthopedic Research Society and the American Society of Mechanical Engineering. In 1998, he was awarded a Whitaker Foundation Research Grant to study human knee joint biomechanics during pedaling using advanced computer simulations. He has authored or co-authored more than 20 papers, about 60 conference papers, and holds editorial positions for several journals.



Michael Aigner is working on his Ph.D. in mechanical engineering at Stanford University. His interests include developing and analyzing mathematical models of complex real-world dynamic control systems, such as human skeletal muscle. He received his undergraduate degrees in mechanical engineering and mathematics from the University of California at Davis and a masters in mechanical engineering from Stanford. 



IPG Team Meets Large-Scale Computing Milestone

To meet its latest milestone, the Information Power Grid team incorporated the updated 512-processor Origin 2000, and ran a 14-million point computation across three remotely located computers.

NASA's Information Power Grid (IPG) team achieved another milestone at the end of December 2000, successfully integrating the updated SGI Origin 2000, *Lomax*, into the grid's infrastructure. *Lomax*, with 512 processors running at 400 megahertz each, is the second large-scale machine to be incorporated into NASA's IPG. After integrating *Lomax* into the IPG production environment, the team demonstrated the grid's capabilities by solving a multi-part aerospace research problem using two additional distributed IPG resources (one each at NASA's Langley and Glenn Research Centers, see Figure 1).

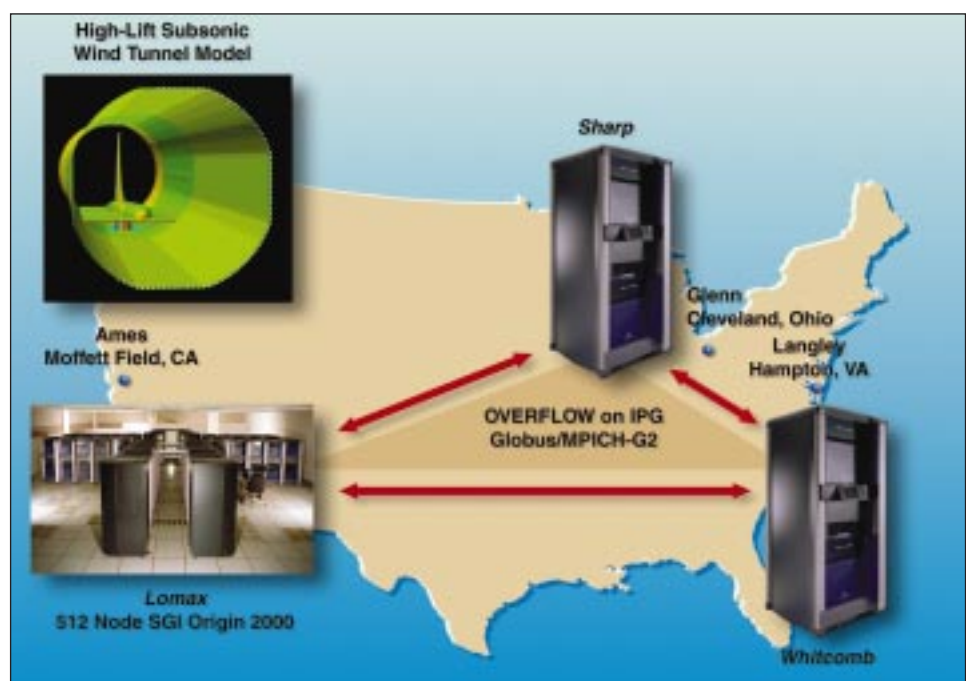
IPG resources *Sharp*, a 24-processor SGI Origin 2000 at Glenn Research Center in Ohio, and *Whitcomb*, an eight-processor SGI Origin 2000 at Langley Research Center in Virginia, were used in concert with *Lomax* to solve complex aerospace geometries using the computer code OVERFLOW, version D, (see sidebar: *OVERFLOW IPG Applications*). This particular version of OVERFLOW is designed specifically to take advantage of parallel and distributed comput-

ing resources by limiting the number of processors that can be used at any one time. "This will enable us to calculate a complex aerospace problem much faster by using all three of the machines," explains Arsi Vaziri, IPG deputy project manager. Using a small number of processors on three distributed machines will reduce turnaround time and create a distributed, collaborative problem solving environment – this approach to computing is also more cost-effective.

The Importance of Data Transfer

In addition to demonstrating integration of a large super-computer into the IPG structure, the December milestone illustrates the importance of data transfer between remote resources. "Our approach to the IPG is geared towards an

Figure 1: *IPG provides aggregated computing in a parallel and distributed fashion. The grid can effectively use underutilized resources, decrease computational turnaround time, support a collaborative problem-solving environment, and provide more cost-effective computing solutions.*



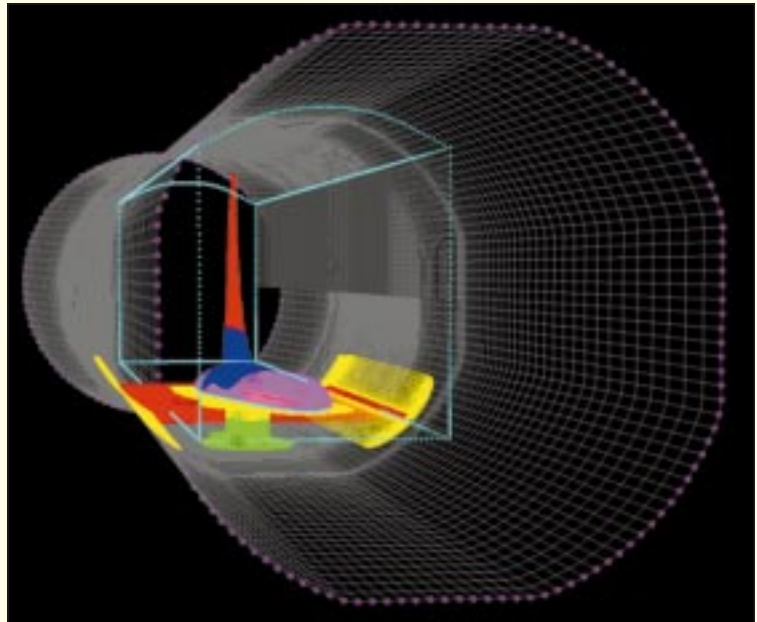
OVERFLOW IPG Applications

One of the large-scale OVERFLOW applications the IPG team used to demonstrate the successful completion of the December milestone was a Navier-Stokes computational fluid dynamics (CFD) simulation of flow about a generic reusable launch vehicle. The simulation was run using a distributed version of OVERFLOW, version D, which required 215 grids, or computational meshes, and approximately 14 million grid points (see figure at right). Without OVERFLOW-D, this application could not have been run on multiple machines. Running this aerospace configuration with the recently integrated 512-processor Origin 2000 verifies that the IPG is capable of supporting aerospace calculations in a production environment.

The team also used a wind tunnel flow simulation of a high-lift wing configuration, which required 32 computational grids and nearly nine million grid points. This type of high-fidelity CFD code provides significant gains for the aerospace industry by reducing design-cycle time.



IPG large-scale application results using OVERFLOW version D. Above: Visualization of a generic reusable launch vehicle X-38 flow simulation – 215 zones and 14 million points. (Maurice Yarow) Below: Wind tunnel flow simulation of a wing-body model encompassing 32 zones and 9 million grid points. (M. Jahed Djomehri)



equal emphasis on data transmission and computational speed,” explains Vaziri. Many of NASA’s newer research programs rely heavily on cross-center collaboration, and IPG provides an economical method for sharing data between colleagues at remote locations.

“Agency-wide programs like ISE (Intelligent Synthesis Environment) are essentially advanced design, or computer design projects for NASA’s aerospace enterprise, and that’s what we’re targeting – a computing environment that enables application domains to work with a large collection of machines, including supercomputers,” explains IPG Project Manager Bill Johnston.

The Globus middleware package, designed to enable different types of IPG resources to interface with each other, was

installed on *Lomax* with very few software changes. “Globus enables the IPG team to easily run any computational code and transfer data between *Lomax*, *Sharp*, and *Whitcomb*,” explains Johnston.

In addition to having the ability to transfer code and data easily, IPG users now have a collection of distributed resources, located at the three NASA centers, to reduce end-to-end turnaround time for aerospace design and simulation problems. The next milestone, scheduled to be met in September 2001, will incorporate an electron microscope located at the University of California at San Diego into the grid’s infrastructure to facilitate experiments requiring high data-transfer rates using the IPG’s high-performance computing systems.



— Holly A. Amundson

ILab: New IPG Tool Improves Parameter Study Implementation

NAS researchers have designed a tool to fully automate parameter study creation and launching within the IPG.

Building large parameter studies may result in a pounding headache – editing hundreds of files, creating shell scripts to run jobs, saving data entered, and shipping the studies off for distributed processing is a recipe for errors. Depending on the number of variables, a parameter study (see sidebar: *Creating Parameter Studies*) can grow very large, very quickly. There is, however, a new software package that can solve this problem. NAS Research Scientist Maurice

Creating Parameter Studies

A parameter study is a collection of computer jobs involving exactly the same computer program, solver, or engineering tool, but each new job is run with slightly different input. For example, aerospace parameter studies typically involve the velocity of the vehicle (Mach number) and angle of attack (in still air, the angle that the nose of the aircraft makes with the direction of travel). Each input file for this aerospace study will have a different Mach number and angle of attack. If there are 10 values for each of the two variables in the study, 100 combinations of Mach number and angle of attack exist, making it a 10-by-10 parameter study. The more combinations, the more complicated a parameter study.

The initial part of the task that ILab automates is the process of creating input files for any number of combinations. Researchers can include any number of varying parameters, creating an n-dimensional parameter study. “I don’t know anybody who wants to use an editor 100 times to create 100 different variations of the same file. It’s time-consuming and prone to error,” says Yarrow.



Yarrow and NAS Software Engineer Karen McCann have been working since early November 1999 to develop ILab, “The Information Power Grid Virtual Laboratory,” a script-generating program. ILab fully automates the process of creating and submitting parameter studies to IPG resources, as well as monitoring them – all within a matter of minutes.

Designed to serve as an IPG research testbed, “ILab is modular and flexible enough so that you can use it to perform IPG parameter study experiments within the context of the tool’s capabilities,” Yarrow says. ILab has been used to test features of Globus, a middleware package designed to enable different applications to interface between IPG resources.

ILab enables IPG users to launch parameter studies and is compatible with many resources connected to the grid. “ILab is very generic, it gives users access to the grid without really having to be a part of the grid. ILab is compatible with Globus although it doesn’t require that researchers have a Globus certificate to use it,” explains Yarrow. “ILab is, in a way, like a gateway to the grid,” adds McCann. The tool will run on any Unix operating system on the network supporting the required level of Perl (Perl 5.005) and Perl/Tk (Tk 8.0). “One of the problems we may face is that not all users will have Perl and Perl/Tk on their systems. We’re trying to work out an installation package to help solve this problem,” explains McCann.

The Ins-and-Outs of ILab

The architecture of ILab can be broken up into a front half and a back half. The front half makes use of a series of graphical user interface (GUI) wizards to collect information about a user’s parameter study, such as resources used to run the job and additional files necessary to complete the study. The

back half of ILab is the code generator – using the information collected by the front half, it generates script files to run the actual parameter study.

Because the scripts are generated automatically by ILab, users have only to answer some very specific questions, prompted by the program. No programming is necessary to run parameter studies using the tool, although users can include their own scripts as ILab “processes.” Currently, there are about 20 screens in the program – all were written using a total of 23,000 lines of Perl and Perl/Tk code. Had the program been written in Java or C++, McCann estimates it would have taken 50,000 to 60,000 lines of code to complete, because she says, these are very “wordy” programming languages.

ILab uses three different methods for sending parameter studies to IPG resources: the local, Globus, and distributed job models. The local job model is the most simplistic, limited to launching small computational jobs in sequential order on a local workstation. The Globus job model (for users with a Globus certificate) sends jobs to remote resources using the Globus middleware package. The distributed job model enables researchers to launch jobs onto remote systems without the requirement of a Globus certificate.

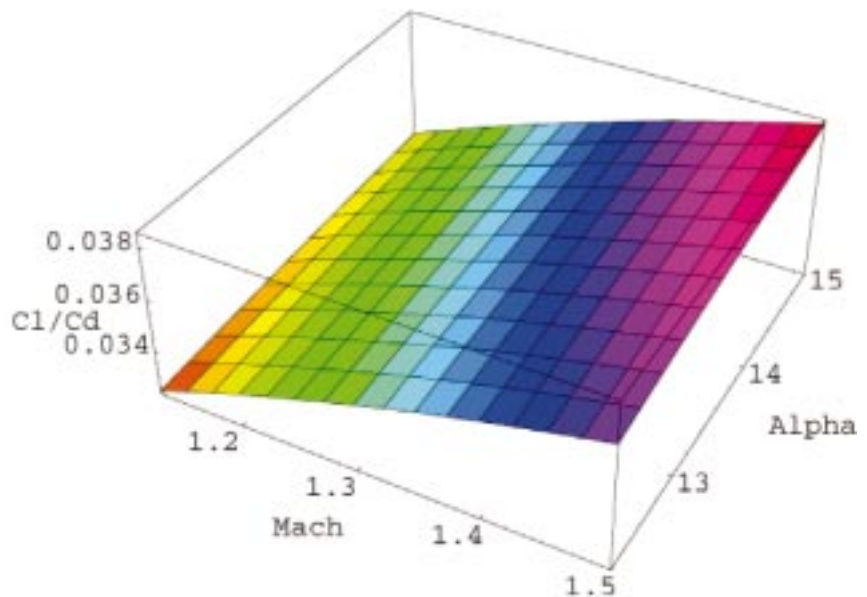
In addition, ILab has the ability to handle a number of other middleware packages commonly found on IPG resources including Message Passing Interface (MPI) and Portable Batch System (PBS) software. “We have designed ILab to be modular in a way that will permit us to easily add new capabilities and make changes to existing capabilities with relative ease,” explains Yarrow. “To make the tool user friendly, we added the ability to enter an arbitrary command, which is necessary because there is always something the user wants to do that we are not going to be able to anticipate,” explains Yarrow. “This way, the user ultimately has flexibility in the event that there is some variety of operation that is not supported by ILab.”

The tool also has a load-balancing algorithm, which interrogates designated machine resources, determines availability,

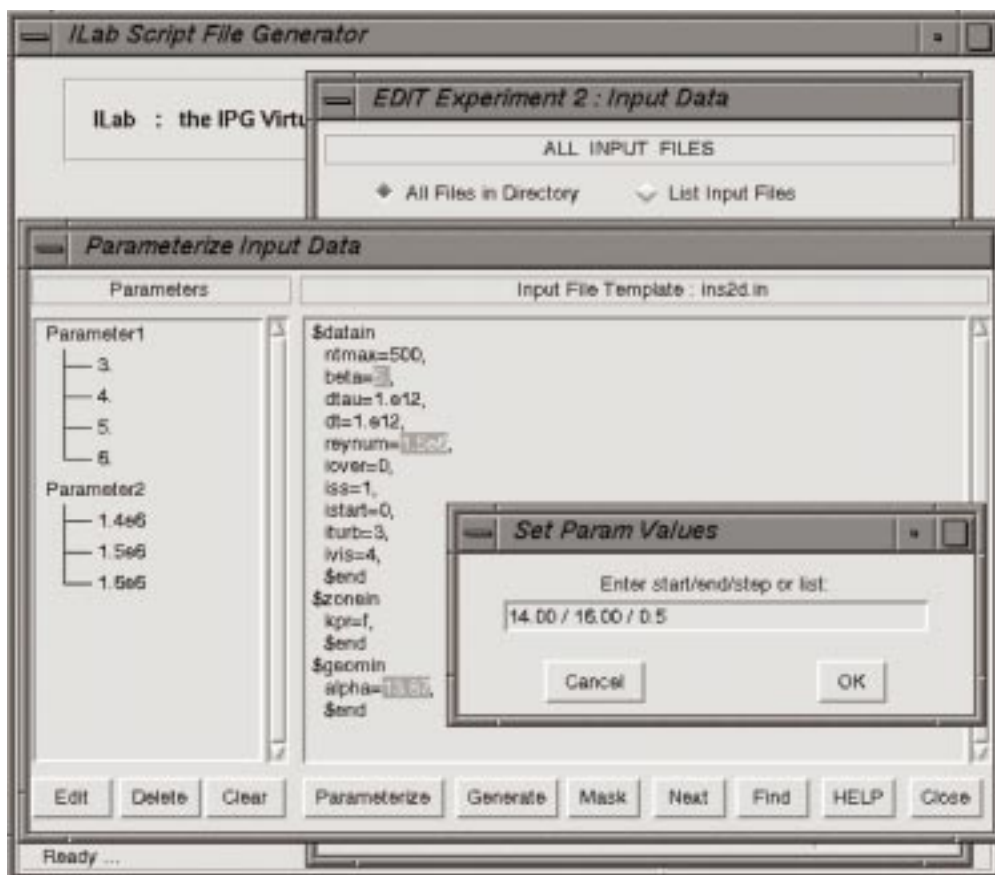


X-38 Crew Return Vehicle with finite-difference mesh system. ILab created and launched a 16-by-12 parametric study in Mach number and angle-of-attack utilizing OVERFLOW-D2 to calculate the flow field around the X-38.

and sends jobs to those machines that are ready for immediate processing. In addition, ILab also has built-in job monitoring capabilities – each job is represented by a set of color-coded icons to help the user determine a job’s progress. McCann and Adrian DeVivo have designed an extensive help package for the tool, complete with a searchable index. “The index is in the form of a hierarchical list. From anywhere in the program users can pop up the help screen – it’s a real user convenience,” explains McCann. ILab also furnishes a list of IPG resources available to researchers for running parameter studies, including detailed information about each machine. In the near future, the team would like to



Coefficients of lift-over-drag for the X-38 at the 192 values of Mach number and angle-of-attack in the ILab parameter study.



Left: ILab's edit parameters screen displays an input file: The user will highlight fields to be parameterized, and then enter a list to create multiple parameterized copies of the input file.

Below: During experiment execution, ILab's job monitor screen shows job status by color-coded icons.

automate this resource list, running automatic resource updates when the ILab application is opened.

Eliminating the Bugs


"There are a lot of possible error conditions; this is why we need to have users test out ILab. Until it has been tested on a significant number of 'real world' problems, you don't really know what error conditions you're going to run into," explains McCann. Users provide Yarrow and McCann with helpful feedback to address errors and inadequacies as they are discovered. Features such as plug-in capabilities, enabling the use of a graphics post-processor of choice, will give users more freedom when setting up parameter studies.

Expanding ILab's Capabilities

Parameter study software is just now starting to surface in the research community. Currently, the size of the user community performing parameter studies is small, but growing. Researchers are beginning to realize that they can find out a lot about their problems by creating multi-variable parameter studies. "ILab is superior to other similar products currently on the market. It has a much more sophisticated user interface, it simplifies the parameterization creation sequence, and it's easy to use because it doesn't require any hand editing of input files," says Yarrow.

Although ILab is still in the pre-alpha testing phase, a great deal of progress has been made on the development of the tool over the past year. Yarrow and McCann plan to contin-



ue work on the tool, testing its capabilities with the assistance of NAS users. Currently, the capability to automatically restart jobs is being added, a feature required by many NAS users performing computational fluid dynamics parameter studies. Improvements like the restart option will be made until the tool is mature enough to handle nearly all parameter studies imaginable. 

— Holly A. Amundson

Space Computing – The Present Frontier

Computer modernization and miniaturization hold the key to longer-duration space missions to the outer reaches of the solar system.

A new generation of computers is on the horizon. Unlike the massive, egomaniacal computer who threatened to take control of the space ship in Stanley Kubrick's *2001: A Space Odyssey*, future space computers will be made of atomic-scale components combined with new methods of computing. Researchers are examining how to design computers with low power requirements that

can withstand the extreme conditions in space. While results of this research could be 15 to 20 years away, planning must begin now.

Computers are essential to NASA's exploration of our solar system. Mission success depends on the reliability and survivability of space computers. While operating on a limited

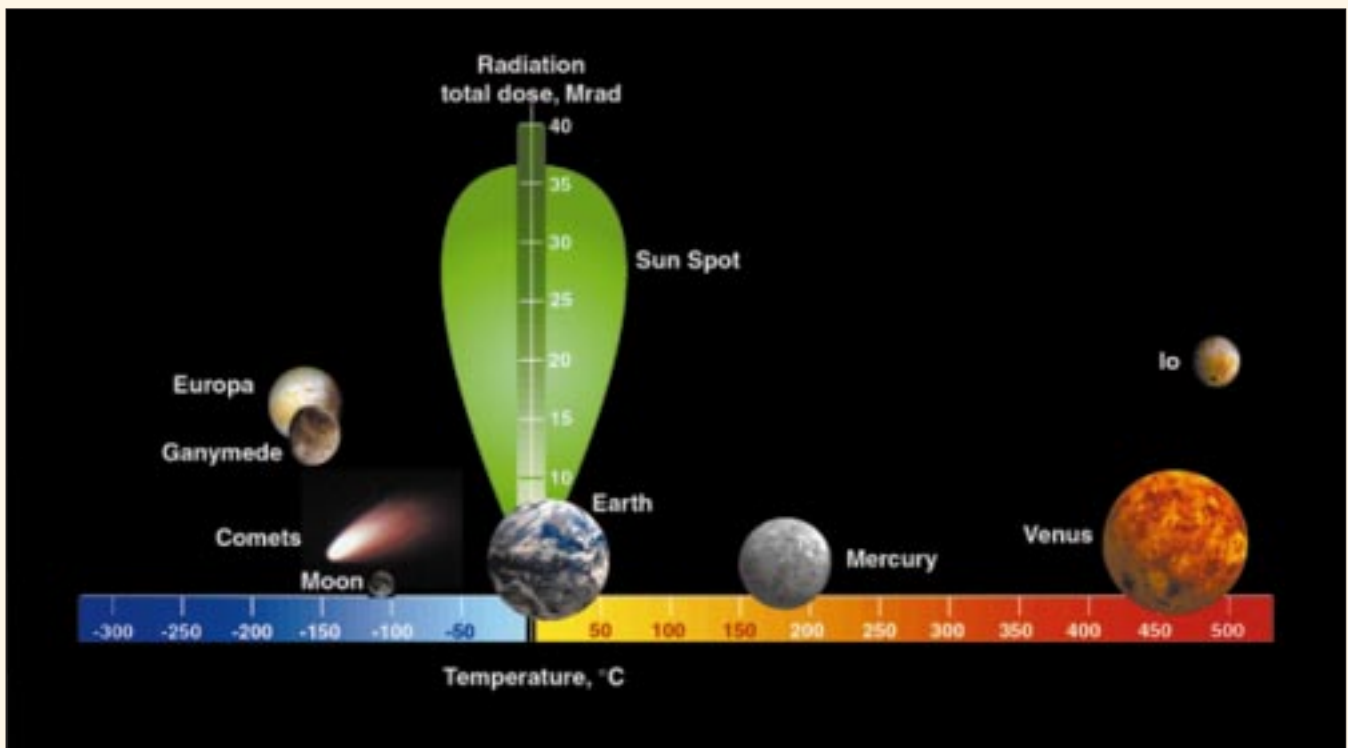


Figure 1: This image represents temperature and radiation levels in the solar system. Earth has one of the mildest environments while the moons of Jupiter (Io, and Europa) represent some of the most extreme conditions of radiation and temperature in the solar system. These moons also generate great scientific interest and are prime targets for exploration. The green shaded region represents the intense burst of radiation from sun spots and is known to damage electronic systems in satellites. (NASA-Jet Propulsion Laboratory).

supply of power, space computers may be exposed to high levels of radiation and extreme temperatures. Computers have performed reasonably well on many NASA missions, despite having less than one-tenth the computing capability of their counterparts on Earth.

While Earth-based computers double in performance every 18 to 24 months, systems used in space have simply been adaptations of terrestrial computer technology. This adaptation to the space environment results in a substantial loss of performance, and does not provide the optimal technology for NASA missions.

To meet the demands placed on computer capabilities for future NASA space missions to Mars, the moons of Jupiter, and other distant planets, new and innovative approaches are necessary. NASA researchers are currently combining new computing models, nanotechnology, and biologically inspired models to address the needs of space computing. Facets of this research will be conducted by the Revolutionary Computing project of NASA's Intelligent Systems (IS) Program.

"The Revolutionary Computing project will look at how space systems are different from Earth-based computing. Space computing has different requirements than Earth computing, such as the need to operate in an extreme thermal and radiation environment, which creates an extra set of challenges," explains NAS Senior Scientist T.R. Govindan.

The project's goal is to increase computer capacity by pursuing new ways of computing that will allow low-power consumption, robust operation under exposure to high levels of radiation and extreme temperatures (see Figure 1, page 15). The IS Program is funded by NASA headquarters and was inaugurated at the beginning of October 2000, merging four areas of computer science research: automated reasoning; human-centered computing; intelligent data understanding; and revolutionary computing.

From Earth to Space

The current trend in ground-based computing is miniaturization – making computer processors as small and as dense as possible. This may not be the best approach for designing space computers because these small, dense processors are less tolerant to radiation and extreme temperatures found in space. A current solution to this problem is radiation hardening, which involves redesigning and repackaging computer chips to negate the effects of radiation. Unfortunately, radiation hardening decreases overall performance, making space-based computers much less capable than their terrestrial counterparts.

"...Revolutionary technologies may address the extreme thermal and radiation environment of space, as well as the limited power and weight requirements for space-based systems."

*–Dan Cooke, program manager
NASA Intelligent Systems*

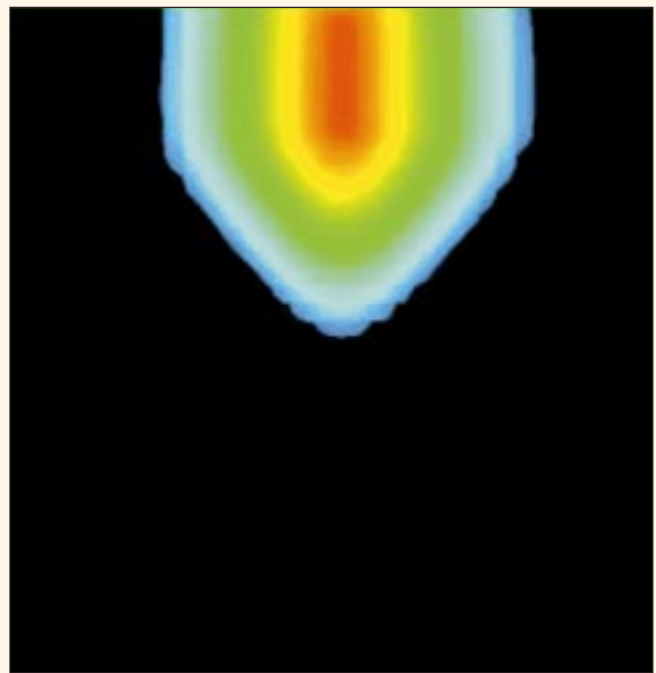


Figure 2: Freeze frame from an Alpha radiation video demonstrating the effects of radiation on a semiconductor in space. The video begins by showing the radiation strike in a confined area on the surface of the electronic device, which eventually spreads throughout the entire semiconductor. Exposure to high levels of radiation can potentially interrupt the operation of a device, threatening the safety of space missions. The use of silicon-on-insulator (SOI) technology is expected to minimize the effect radiation has on electronic devices by keeping a majority of the generated electric charge away from active systems in space. (Bryan Biegel)

When an electronic device is exposed to the high levels of radiation found in space, the entire system can be damaged, rendering it useless (see Figure 2). Much progress has been made in silicon-on-insulator (SOI) technology devices – most recently with the creation of an insulating layer to prevent radiation from reaching active regions of an electronic device and destroying it. Unfortunately, SOI devices tend to have self-heating effects and do not protect well against all types of radiation in space.

"Recent results in physics and biology may lead to revolutionary approaches to computer hardware," says Dan Cooke, IS program manager. "These revolutionary technologies may address the extreme thermal and radiation environment of space, as well as the limited power and weight requirements for space-based systems. The Revolutionary Computing Program is designed to position NASA so that it can better exploit these new technologies should they come to full fruition."

“We are focusing on the computational models and languages implied by these new approaches, and the development of certain massively parallel algorithms that will be of benefit to NASA in the future. Results from the program may also be beneficial to the development of problem solutions that exploit more conventional approaches to computing,” Cooke says.

Innovative Approaches

One innovative approach to improving space computers is to mimic the design of biological systems. “In general, biological processes are more fault-tolerant and they degrade gracefully – if some part of the device is damaged, it doesn’t sabotage the entire system,” explains Govindan.

With traditional computing, if some part of the processor is damaged, the whole computer fails. Neural computing, for example, may potentially provide a superior model for complex information processing and problem solving. Neural networks contain a large number of simple interconnected and interacting neurons – a helpful model for designing space systems capable of exploring a massive amount of data in a short period of time. Current work with neural computing deals with instantaneous recognition of patterns; an example is face recognition in a static image. Future research will deal with dynamic, or moving environments. Another approach to solving computationally complex prob-

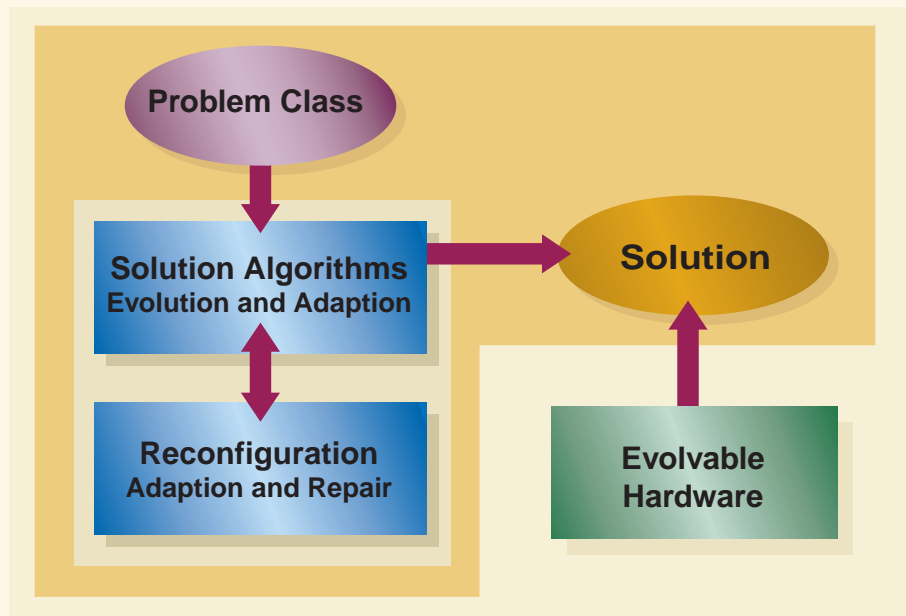



Figure 3: *This evolvable computing model uses genetic and evolutionary algorithms to solve complex computing problems. Researchers working on the Revolutionary Computing project are currently looking at ways to solve problems using biologically inspired algorithms modeled after selective reproduction, mutation, and genome crossover to search for an optimal solution. (Jason Lohn)*

lems rooted in biology is to use genetic and evolutionary algorithms (See *Gridpoints*, Winter 1999, page 4). Genetic and evolutionary processes naturally occurring in biological structures such as DNA, cell functioning, and ecosystems are excellent models for creating programs designed to manage vast amounts of data and arrive at an optimal solution (see Figure 3). Using biologically inspired models does, however, create a challenge – data must be translated in a way suitable for arriving at a reasonable solution.

Nanotechnology, the manipulation of materials and devices at the atomic- and nanoscales will also help improve space computing. Nanotechnology devices consume less power, and combined with architecture and new models of computation could be more robust in the extreme radiation and temperature environment of space. NAS researchers in the science and technology group are examining what new types of devices can be made, and which ones would be suitable for space computing applications.

New models of computation, combined with new materials and devices could be the answer to current problems with computers used to conduct missions to distant planets. Future space missions include exploration of Io and Europa; however, planning must start now in order to make the journey into deep space possible. 

—Holly A. Amundson



Artist's rendering of a hydrobot exploring oceans beneath Europa's icecap. A mission of this type would present extreme environments for computing. (NASA/Jet Propulsion Laboratory)

Chimera Grid Tools Software

A new, updated version of the comprehensive Chimera Grid Tools software package for overset grid generation has recently been released.

By William M. Chan and Stuart E. Rogers

In computational simulations of flows about an object, a computational grid is used to model the object's geometry. For configurations involving complex geometry, viscous flow physics, and bodies in dynamic motion relative to each other, the Chimera overset grid method is currently one of the most computationally cost-effective options for obtaining accurate aerodynamics results. Considerable success has been achieved in applying this method to a wide variety of problems, as shown on page 19.

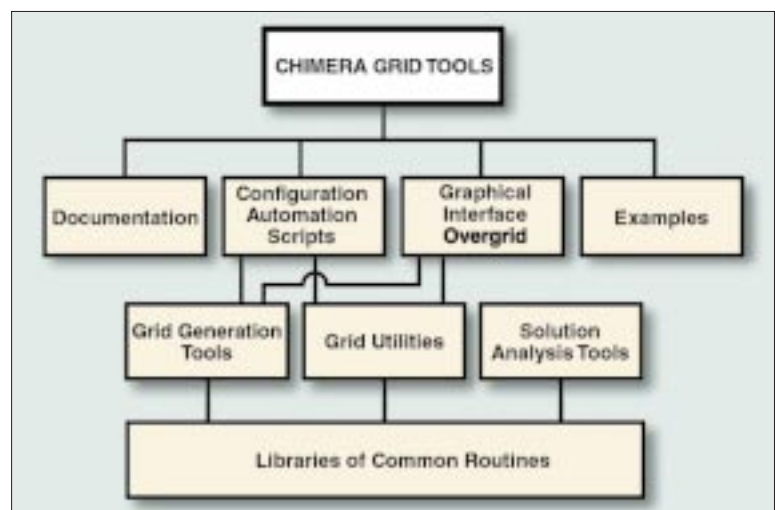
Chimera Grid Tools (CGT) is a software package containing a suite of tools for creating overset computational grids, and performing geometry processing, grid diagnostics, solution analysis, and flow solver input preparation. Scientists from the Ames Army/NASA Rotorcraft Division and NAS Systems Division released CGT version 1.4 in October 2000. Since its inception at NASA four years ago, CGT has been requested by and distributed to more than 200 domestic, commercial, and U.S. government organizations under non-disclosure agreements. The software has been utilized in aerospace, marine, automotive, environmental, and sports applications. In addition to the authors, Steve Nash (formerly with MCAT Inc. at NASA Ames) and Pieter Buning (NASA Langley) collaborated on the development of CGT.

"Engineers in Aerodynamics Technology at the Boeing Co. (Long Beach, Calif.), routinely use CGT and the OVERGRID GUI (graphical user interface) for overset grid generation on a wide variety of configurations," says Jeff Slotnick, aerodynamics technology principal engineer. "Examples of recent projects using Chimera Grid Tools and the OVERGRID GUI include commercial and mili-

tary transports, the Blended Wing Body, and Advanced Theatre Transport configurations."

"We use the Chimera Grid Tools on a daily basis to support a wide range of Space Shuttle and International Space Station analyses," says Reynaldo Gomez, a senior engineer at NASA Johnson Space Center. "CGT is rapidly becoming a comprehensive set of tools for generating and manipulating overset grids. One of the main benefits of this set of tools is that it collects utilities developed by the overset community into a single package that runs on a wide range of platforms."

The current procedure for performing computational flow analysis on complex configurations typically involves the following four steps: Geometry processing to obtain a clean



Chimera Grid Tool's hierarchical structure: low level library routines are utilized by individual batch modules which themselves are called by the configuration scripts and the graphical interface.

description of the configuration; grid generation to obtain a set of overset composite grids around the configuration; flow solution computation at given flow conditions; and post-processing to obtain integrated forces and moments, local pressure profiles, and various other flow features data.

The Structure of CGT

CGT consists of a hierarchy of software modules together with documentation

and examples (see diagram, page 18). At the lowest level are libraries containing commonly used functions such as input/output routines for data files, stretching functions, projection routines, and many others. This set of libraries is gradually evolving into an applications programming interface (API) for grid-generation utilities.

At one level above the libraries are about 30 independent programs that can be used in batch mode. These programs have been developed individually, are continuously updated, and range from a few months old to about 12 years old. Capabilities offered by these programs include editing, redistribution, smoothing and projection of grids, hyperbolic and algebraic surface and volume grid generation, and Cartesian grid generation. At the highest level, two pieces of software communicate with the individual programs. The first is a graphical user interface and the second is a collection of configuration automation scripts.

The OVERGRID Graphical User Interface

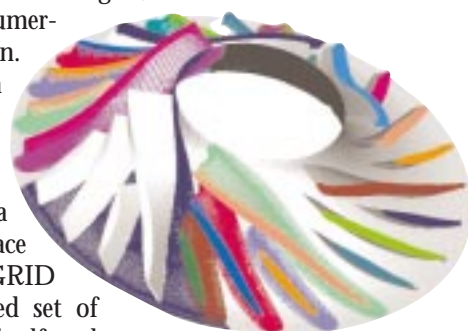
The primary function of OVERGRID is to provide a single graphical interface environment for performing a wide range of operations prior to running the flow solver. (See: www.nas.nasa.gov/~rogers/cgt/doc/overgrid.html). These operations typically include checking and processing the input geometry, generating the surface and volume grids, analyzing the grid quality using various diagnostics, and creating input parameter files for the domain connectivity program and the flow solver. Currently, OVERGRID can be used to create structured grids from a geometry description composed of multiple panel networks or triangles. Users of OVERGRID have reported a reduction in grid generation time by as much as a factor of four.

A new module recently added to OVER-

GRID assists users in preparing inputs for the OVERFLOW and OVERFLOW-D flow solvers. For dynamic simulations involving bodies in relative motion, the OVERFLOW-D input file may require up to 100 global parameters and about 70 to 90 parameters for each grid,

depending on the numerical scheme chosen. Preparing such an input file for a complex configuration can be a formidable task, especially for a new user. The interface offered in OVERGRID

provides a simplified set of commonly used and self-explanatory options for the user to choose from. A fictitious example using the Space Shuttle Launch Vehicle illustrates the type of simulations that can be prepared using the new OVERFLOW-D interface in OVERGRID (see Figure 1, page 20).

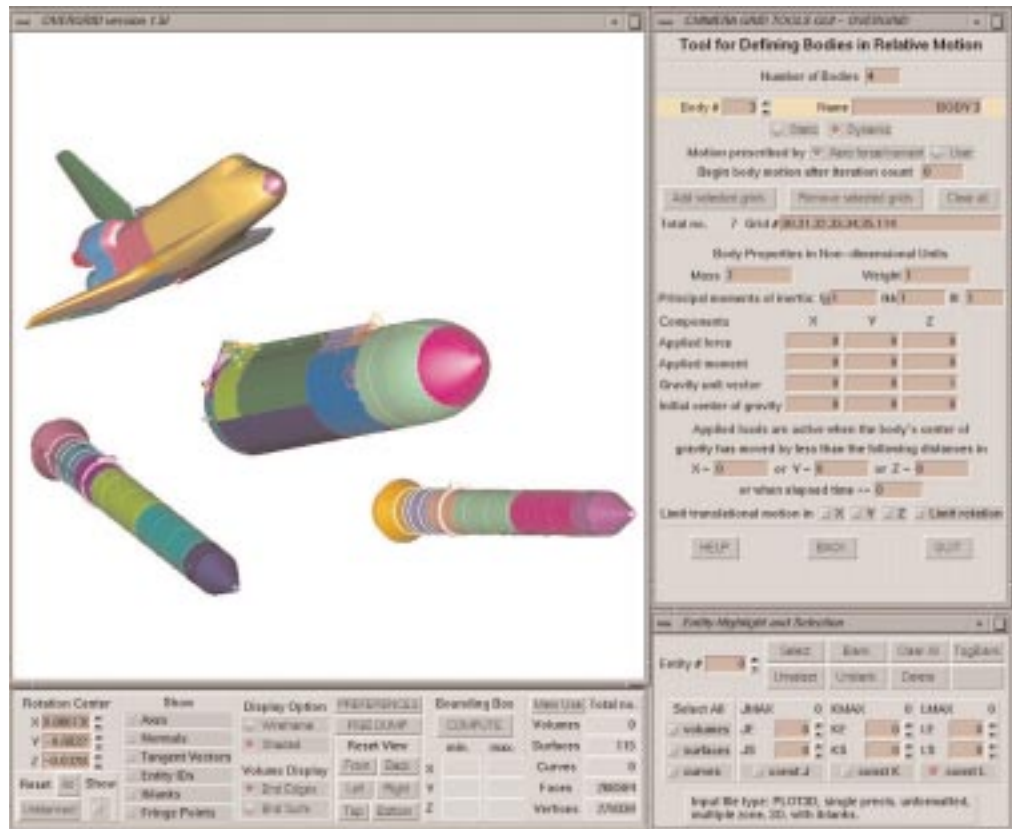


Configuration Automation Scripts

In a typical computational fluid dynamics (CFD) analysis, many different CGT programs are used to proceed through the grid generation phase. Most of these tools require similar information about the grid system, including solid wall information, boundary condition information, and information about which surfaces form separate components in the configuration.

A sample of recent simulations using grids created by Chimera Grid Tools (clockwise from top left): Surface grids on a Navy submarine (Applied Physics Laboratory, Johns Hopkins University); Impeller section of the Space Shuttle Main Engine (NAS Systems Division, NASA Ames Research Center); V-22 tiltrotor (Army Aeroflightdynamics Directorate); and the X-38 prior to release from the NASA B-52 airplane (NASA Johnson Space Center).

Figure 1: An example of a fictitious Space Shuttle Launch Vehicle separation procedure showing the OVERGRID-OVERFLOW-D interface for creating inputs for a simulation involving four bodies in dynamic relative motion. Each body may contain multiple grids where each grid is depicted in a different color. Simple widgets allow the user to define the different bodies and to specify their attributes such as mass, moments of inertia, center of gravity and applied loads.



A Tcl (Tool command language) script system was recently added to the CGT package that provides automation for many of the steps performed during the grid generation process. The scripts were developed in support of work in the Advanced Subsonics Technology Program, where a group of CFD specialists were tasked with computing the flow over a Boeing 777 in a landing configuration in less than 50 working days. The geometry and surface grids for this configuration are shown in Figure 2. The group was able to exceed the program's milestone by computing the first solution of this high-lift aircraft in only 32 working days.

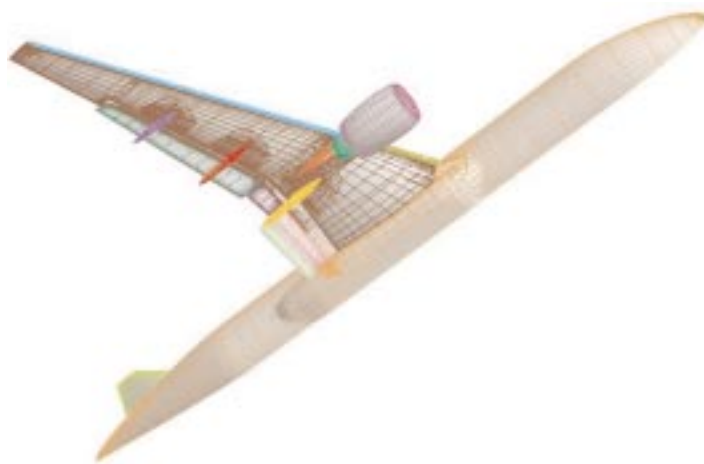



Figure 2: Surface grids on a Boeing 777-200 transport aircraft in landing configuration, which was analyzed using the CGT script system.

User Input Improves Product

CGT is being continuously updated and improved. New features and capabilities are constantly added, usually as a result of requests from the large user community. After the release of version 1.4, input from users will be compiled and incorporated into a future release of CGT. Feedback from real-world users of CGT has enabled the software program to evolve into the robust design tool that it is today. More information on CGT is available at: www.nas.nasa.gov/~rogers/cgt/doc/man.html. 

About the Authors

William M. Chan is currently a senior research scientist at ELORET in the NAS Systems Division's Applications Branch. He is the primary author of OVERGRID and many of the modules in CGT. Since 1990, he has been a contractor at NASA Ames working on algorithm and software development for over-set grid technology in support of the Space Shuttle, Advanced Subsonic Technology, and Rotorcraft programs.

Stuart E. Rogers is an aerospace engineer also working in the Applications Branch of the NAS Systems Division. He has been working at NASA Ames since 1985 on the development and application of CFD software. His work has included the co-development of the INS2D and INS3D flow solvers, Chimera Grid Tools, and most recently, further development of the over-set-grid software known as PEGASUS. His applications have included problems in high-lift aerodynamics, bio-fluid flows, and hydrodynamics.

Calendar of Events

IEEE-NASA Symposium on Mass Storage Systems

San Diego, California • April 17-20

The 18th IEEE Symposium on Mass Storage Systems and 9th NASA Goddard Space Flight Center Conference on Mass Storage will focus on the issues surrounding the storage of large data collections that are web accessible. Topics for discussion include managing large volumes of data, long-term mass storage requirements, and experiences in fielding solutions. Emphasis is on current and future practical solutions addressing issues in data management, storage systems and media, data acquisition, long-term retention of data, and data distribution. Conference information is available at: www.storageconference.org/2001

International Parallel and Distributed Processing Symposium (IPDPS 2001)

San Francisco, California • April 23-27

IPDPS is an international forum for engineers and scientists from around the world to present their latest research findings in all aspects of parallel computation. In addition to technical sessions of submitted paper presentations, the meeting offers workshops, tutorials, an industrial track, and commercial exhibits. Simultaneously, the Workshop on Parallel and Distributed Scientific and Engineering Computing with Applications (PDSECA-01) will also be held at the same venue. Visit www.ipdps.org for more information.

The 2001 International Conference on Computational Science

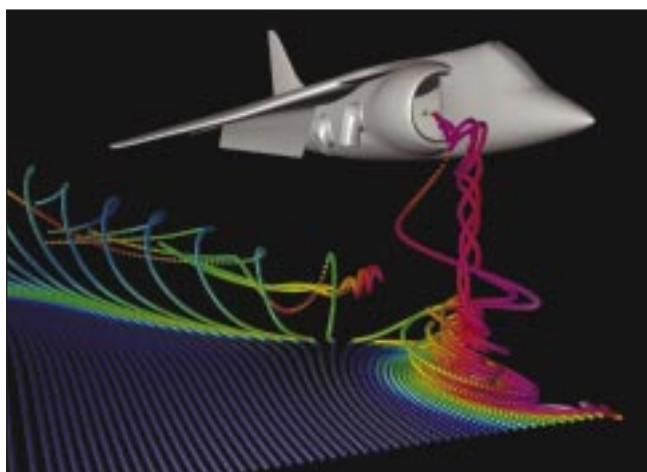
San Francisco, California • May 28-30

The 2001 International Conference on Computational Science is becoming a vital part of many scientific investigations, affecting researchers and practitioners in areas ranging from aerospace and automotive, to chemistry, electronics, geosciences, to mathematics, and physics. Attendance information is available at: www.ucalgary.ca/iccs or visit: www.hpcc.rdg.ac.uk/iccs

Mobile Terrestrial and Space Networking Workshop

NASA Ames Research Center • June 25-27

Hosted by NASA Research and Education Network (NREN) and sponsored by NREN and the National Science Foundation (NSF). The objective of this workshop is to identify and explore various high-end wireless, mobile, and satellite communications networking technologies, as they can be used to support the scientific community. The



Visualization of a Harrier vertical take-off and landing fighter jet with a 33-knot headwind, hovering 10 feet above the ground. The engine inlet is ingesting a ground vortex. Computation by Jasim Ahmad, Neal Chaderjian, Scott Murman, and Shishir Pandya. Visualization by Tim Sandstrom of the NAS data analysis group.

technical program will be divided into three themes: technologies and architectures, seamless integration of heterogeneous networks, and applications. Note that participation is by invitation only. For more information, contact Marjory Johnson at mjjohnson@arc.nasa.gov.

Sixth Annual Computational Aerospace Sciences Workshop

NASA Ames Research Center • July 31—August 2

NASA's High Performance Computing and Communications Program is hosting its sixth annual technology forum on Computational Aerospace Sciences (CAS) at NASA Ames Research Center, Silicon Valley, Calif. Researchers from NASA, industry, academia, as well as other government agencies will discuss the challenges and successes of their work. For further details, visit: www.cas.nasa.gov/cas2001.htm

10th IEEE International Symposium on High Performance Distributed Computing

San Francisco, California • August 7-9

The 10th International Symposium on High Performance Distributed Computing is a forum for presenting the latest research findings on the use of networked systems for high-performance computing. Submissions are encouraged on all aspects of high-performance distributed computing, visualization, and collaboration, including hardware technologies, network protocols, the middleware that ties distributed resources together into "computational, data, and collaboration Grids," middleware that enables application use of Grids, and tools and languages that support application development. Detailed information is available at:

www.itg.lbl.gov/HPDC-10/ 